Hindawi
Security and Communication Networks
Volume 2021, Article ID 6661954, 8 pages
https://doi.org/10.1155/2021/6661954

Research Article

Machine Learning-Based Stealing Attack of the Temperature Monitoring System for the Energy Internet of Things

Qiong Li,[1] Liqiang Zhang,[2] Rui Zhou,[2] Yaowen Xia,[1] Wenfeng Gao ,[1] and Yonghang Tai [2]

[1]*Solar Energy Research Institute, Yunnan Normal University, Kunming, Yunnan 650500, China*
[2]*Yunnan Key Laboratory of Opto-electronic Information Technology, Yunnan Normal University, Kunming 650000, China*

Correspondence should be addressed to Wenfeng Gao; 413900096@qq.com and Yonghang Tai; taiyonghang@126.com

Received 18 December 2020; Revised 31 January 2021; Accepted 9 March 2021; Published 25 March 2021

Academic Editor: Weizhi Meng

With the development of the Energy Internet of Things (EIoT), it is of great practical significance to study the security strategy and intelligent control system for solar thermal utilization system to optimize the operation efficiency and carry out intelligent dynamic adjustment. For buildings integrated with solar water heating systems, computational fluid dynamics simulation was used in analyzing the process of solar energy output. A method based on machine learning is proposed to predict energy conversion. Besides, the simulation and analysis are carried out in combination with the possible safety problems such as the vibration of the control system. This paper proposed a novel platform of EIoT for machine learning-based cybersecurity study and implemented the platform for the temperature monitoring system. After the evaluation of the machine learning-based cybersecurity study, the EIoT system demonstrated a high performance with the Extreme Gradient Boosting (XGBoost) training algorithm.

1. Introduction

With the resonant coupling of multiple energy sources, the new energy supply system under the background of the Internet of Things will have the characteristics of multi-energy complementation and synergy [1–3]. The traditional thermal and electric demand response will gradually develop into a comprehensive demand response suitable for integrated energy systems. The EIoT (Energy Internet of Things) has brought a better operation, monitoring, and management mode for the new energy utilization systems. It uses advanced sensors, control, and software applications to connect a large number of equipment, machines, and systems at the energy production side, energy transmission side, and energy consumption side to form the "Internet of Things foundation" [4]. Big data analysis, machine learning, and prediction are the important technical support for the EIoT to realize the life characteristics. Therefore, in the solar water heating system, the sensor measurement technology, numerical simulation technology, Internet of Things technology, and machine learning technology can be used to remote monitoring of the control parameters, heat gain, and status of the solar water heating engineering system, so as to realize the saving, comfortable, efficient and reliable water, energy and heat consumption, which has strong practical application value [5].

However, in the operation environment of EIoT, the operation control of a high-proportion new energy system highly relies on low-latency and highly reliable information and communication technology, which brings excellent technical challenges to the system's network security defense.

In the applications of IoT, in recent years, some critical solutions have been put forward by many experts and scholars to security issues. Xu [6] listed some security problems and critical technologies of IoT. Mahmoud et al. [7] provide eight frameworks for the security applications in the IoT field, each of the framework addresses in detail the security measures and the ability to fight against the attacks. In each specific application scenario, faced with the security demand of the smart home environment, Huichen Lin and Neil Bergmann [8] proposed two critical technologies of

auxiliary management to deal with these problems. Minoni et al. [9] provided relevant tools and technologies to address security vulnerabilities in e-health and assisted living applications. Baranwal et al. [10] designed a remote control and monitoring device; it can be used for the security detection of farmland, grain storage, and cold storage. At the same time, ML can provide algorithm support for the designed whole IoT system. For example, when we develop a system that uses text data for classification, we can adopt the ML method to process the data and set our system according to the actual situation. The researchers above have given us an idea to use ML to design a new platform for the temperature monitoring system. The system is equipped with high security and privacy and can be applied to daily life.

(1) Create a new platform of EIoT for machine learning-based cybersecurity study

(2) Propose a model stealing attack on the intelligent energy supply system

(3) Implement the proposed intelligent energy supply platform and model stealing attack

The structure and content of this paper are organized as follows: in Section 2, we review the related works on the cyber-attacks with machine learning for the EIoT. The model stealing attack experiments are designed in the methodology part, which is presented in Section 3. In the next section, the performance of attacks on the medical platform was demonstrated and discussed. In the last section, we summarize the results and conclude this paper.

2. Related Work

Another algorithm in machine learning, the RandomForest algorithm, also is used in systems of IoT. In the face of IoT security problems, Nawir et al. [11] directly summarized various attacks into a well-structured classification to help researchers and developers, so that security measures can be planned appropriately in the development of the IoT. Overall, with the extensive application in IoT of various fields, the security of IoT will always be a hot research direction. However, sometimes existing solutions are insufficient to cover the security range of IoT; machine learning (ML) technology can provide embedded intelligence in IoT devices and networks to address security issues [12]. According to the unique characteristics of IoT devices and environments, Zeadally et al. [13] mentioned the relevant advantages of ML algorithms. Gomes et al. [14] used this set of an algorithm for indoor positioning of users in a smart home. As a basic algorithm in machine learning, the XGBoost algorithm is applied to a wide range of modern industries. The main advantage of XGBoost is its scalability, and speed of execution is usually superior to other ML models [15]. When dealing with classification problems, compared to other models, XGBoost has better classification accuracy [16]. In the field of IoT, sometimes XGBoost is used as a method to detect if a system has been compromised [17]. In this paper, XGBoost has been applied to solve our data classification; on this basis, our system is also equipped with high security.

3. New Platform for Mobile and Intelligent Medicine

3.1. EIoT System Design

3.1.1. Machine Learning Models for Monitoring Water Instantaneous Flow. In this article, we use RandomForest to classify the water temperature measured from the water tank outlet and then use XGBoost to steal the entire network. RandomForest is a classifier containing multiple decision trees, and its output category is determined by the model number of the categories output by individual trees. It was first proposed by Leo Breiman and Adele Cutler. We can think of a decision tree as a collection of if-then rules. Decision tree learning can be described by $Pi = Xi/M,\ i = 1, 2, \ldots, N$, in which x is the input instance (eigenvector), M is the number of features, and i is the class tag, $i = 1, 2, \ldots, N$. N is the sample size [breiman] RF constructs bagging integration on a decision tree-based learner and further introduces random attribute selection in the training process of the decision tree. XGBoost is a tree integration model. Assuming that there are k trees, so the sum of the predicted values of each of the k trees for the sample is used as the prediction of the XGBoost model.

Given a dataset, including z samples and s features, $\mathscr{T} = \{(x_i, y_i)\}$ $(|\mathscr{T}| = z, x_i \epsilon \mathscr{R}^s, y_i \epsilon \mathscr{R})$.

The output of the tree model is

$$\widehat{y}_i = \varnothing(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \varepsilon \mathscr{F}. \tag{1}$$

The space of CART tree is F, as follows:

$$\mathscr{F} = \{f(x) = \omega_{q(x)}\}\left(q: \mathscr{R}^s \longrightarrow H, \omega \varepsilon \mathscr{R}^H\right), \tag{2}$$

where q represents the model of the tree. Input a sample and map the selection to the leaf node according to the model to output the predicted score. $\omega_{q(x)}$ represents the set of fractions of all leaf nodes of tree Q; H is the number of leaves in the tree q. Therefore, it can be seen from equation (1) that the predicted value of XGBoost is the sum of the predicted values of each tree; namely, the sum of the scores of the corresponding leaf nodes of each tree (ω) i represents the score of the ith leaf node. Our goal is to learn K tree models like this $f(x)$. First, define a target function:

$$\mathscr{L}(\phi) = \sum_i l(\widehat{y}_i, y_i) + \sum_k \Omega(f_k), \text{ where } \Omega(f) = \gamma T + \frac{1}{2}\lambda\omega^2. \tag{3}$$

The optimization parameter of the XGBoost model is model F (x), rather than an additive value, so we cannot use traditional optimization methods to optimize the Euclidean distance. We use additive training to learn the model.

3.2. Model Stealing Attack to the New Platform

3.2.1. Overview of the Threat Model. As we can see in Figure 1, the IoT temperature control system is divided into three different layers by us. In the player, we focus on collecting the functions that users need to implement in this system. In this article, it is precisely the temperature monitoring of the water tank outlet; the second layer is the *n* layer, focusing on modelling the entire user requirements. In this article, the RandomForest algorithm is mainly used to monitor the temperature of the water tank, and after experiencing our system, users will provide suggestions, and then we make improvements to our system based on these suggestions. This layer is to feed back user opinions to us so that our network can better meet customer needs.

Dividing the IoT network into three different layers is the result of consideration from the perspective of network security. Doing so can reduce the risk of network attacks on the entire system. In the perceptual layer, Ian et al. [18] proposed the concept of input, which forms input by imposing small but deliberate worst-case perturbations on the examples in the dataset, so that they can output a high-confidence wrong answer.

For the solar hot water monitoring platform, its main purpose is to monitor and effectively dispatch the relevant data in the solar hot water system, such as monitoring the water level data of the water tank, and setting different water replenishment strategies according to the change of water level; or setting different water temperature heating strategies, so that the solar hot water system with insufficient sunlight can heat up automatically in time. Therefore, the system needs to effectively receive the data from the data acquisition end and establish the corresponding database, so as to obtain the corresponding data change curve, such as the water temperature change curve of the water tank, so as to facilitate the subsequent prediction of the water temperature change and set the corresponding maintenance strategy.

3.2.2. Theoretical Description of the Model Stealing Attack. We will introduce how to use data stolen from an existing target network (RandomForest in this case) to build our copycat network, in this part. The importance of the whole process is to use random natural data to instruct a network of imitators from the existing target network. Two steps have been included: the creation of pseudo training data is used to train the imitator network. First, use the target network as a grey box to label random natural data to generate a pseudo dataset. Then, use the pseudo dataset to train the simulated network and copy the attributes of the target network (Figure 2).

In Figure 2, we briefly explained how to build a copycat network. Now, we will introduce this process in detail. Corresponding to Figure 2, the entire copycat network training process should be divided into two parts. The first is an essential training set. The training set used by copycat builds and the training set used by the target network are in the same problem [19–21]. The point that needs to be emphasized here is that although they use data in the same

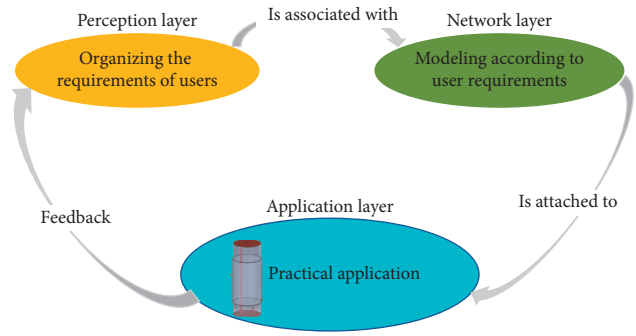

Figure 1: EIoT system framework based on the temperature control system.

problem domain, their datasets are not the same, because random sampling is generally used when collecting data. If there is a large amount of data duplication when adopting the copycat dataset, we can also go online and use natural data to augment the copycat dataset, so as to avoid copycat and the target network using the same dataset and affecting the final result judgment error. Importing the selected dataset into the original professional network to steal the corresponding labels is the most critical step for the entire copycat network. At this time, the quality of the titles stolen will often determine whether the copycat network can fully implement the original network.

After using the copycat dataset to steal the appropriate labels from the target network, the next step is to use these datasets to train the selected network. In this article, we chose the XGBoost *t* as the attack model. The reason for the choice is that it is the same machine learning algorithm as RandomForest, but there are also differences: RandomForest processes data in serial, while XGBoost processes data [22]. It is parallel processing. This choice of the network also proves the feasibility of our network attack from the side. During training, we imported the copycat dataset and its corresponding labels generated by ourselves into the XGBoost network and imported the same test set as the original network to determine whether the copycat network we created can achieve the accuracy of the average temperature and abnormal temperature in the input data classification.

Next, we will explain the assignability of adversarial samples. Suppose that the adversary is interested in classifying the wrong sample and producing a hostile sample $\overrightarrow{\omega^*}$ different from the model in which the class is assigned to the legal input $\overrightarrow{\omega}$. In the following optimization formula, we can achieve this:

$$\overrightarrow{\omega^*} = \overrightarrow{\omega} + \theta_{\overrightarrow{\omega}} \text{ where } \theta_{\overrightarrow{\omega}} = \arg\min_{\overrightarrow{\alpha}} g(\overrightarrow{\omega} + \overrightarrow{\alpha}) \neq g(\overrightarrow{\omega}). \tag{4}$$

$\overrightarrow{\omega^*}$ is the hostile sample, and g is the activation function. However, adversarial samples are often incorrectly classified as g' instead of g in practice. For the convenience of discussion, the concept of transferability of adversarial samples is formalized [23]:

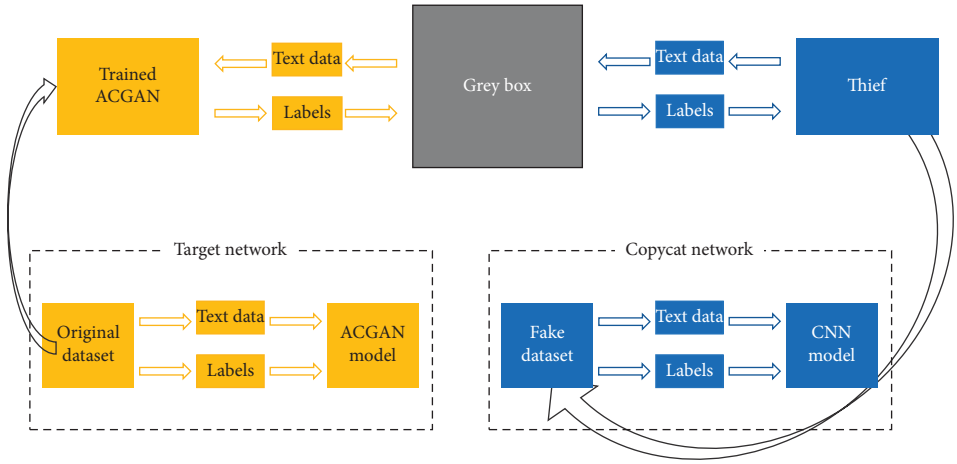

Figure 2: The figure shows how to steal a model from a trained network. The entire network can be divided into two parts; that is, copycat is divided into two parts. First of all, we see that the part on the left is the existing network, which is what we call the target network. It uses the data we already have for training and testing, while the network on the right is the copycat network. Its basics are that it is based on a certain cognition of the target network, and its training set and the training set used by the original network are in the same problem domain.

$$\Pi_Y(g, g') = \left|\left\{g'(\vec{\omega}) \neq g'\left(\vec{\omega} + \overrightarrow{\theta_{\vec{\omega}}}: \vec{\omega} \in Y\right)\right\}\right|. \quad (5)$$

Set Y represents expected input distribution solved by the models g and g' in the task. We divide the transferability of adversarial samples into two variables to describe the models (g, g'). The first is the transferability within the technology, the transferability between different parameter initializations of the same technology or training models of different datasets (for example, g and g' are both deep learning networks or both support vector machines (SVM)) is defined by which. Second, for cross-technology transferability, two technologies can be used to train models (for example, g is a deep learning network and g' is SVM).

4. Experiments and Results

4.1. Implementation of the Platform

4.1.1. Discussion on Specific Temperature Control System Usage Scenarios and the Attack. The solar heating system is the most widely used solar energy utilization system. The dynamic modelling, characteristic analysis, and optimal control of solar heating systems play an essential role in promoting intelligent applications, safety, and convenience. The numerical calculation method of the dynamic thermal characteristics of the solar heating system is an effective means for the thermal dynamic modelling and analysis of the heating pipe network. However, the time and space complexity of the numerical calculation method are relatively large, and it is necessary to deeply analyze the time and space of the numerical calculation method. Due to its excellent algorithm, machine learning can optimize the numerical calculation performance of the thermal dynamics of the heating system, thereby providing a basis for the rapid analysis and optimization of the non-steady thermal process of an extensive heating network. Moreover, when the heating network performs frequent and wide-range temperature and flows adjustments, it may cause the control system to oscillate, which in turn causes the dynamic instability and imbalance of the heating network, so that our temperature control system has its place.

The intelligent solar heating control system based on EIoT can be seen in Figure 3. Energy supply, storage and transfer, energy management system, and energy consumption have been contained within the system. The working process of the whole system can be described as follows: after the solar energy is captured by the energy acquisition system, the energy is converted into thermal energy for storage by heating cold water. After the energy monitoring system determines that the output water temperature reaches the safe water temperature, the energy will be output to the user's home.

4.1.2. Discussion on Specific Temperature Control System Usage Scenarios and the Attack. In this section, a solar water heating system for buildings is studied, and its hot water output process is simulated by computational fluid dynamics (CFD). Considering the actual operation of the solar balcony wall hanging system, the inlet temperature of the circulating working medium is set as 338 K, and the initial water temperature inside the water tank is 288 K. The temperature contours and velocity vector diagram change with time when the mass flow rate of the circulating working medium is 0.022 kg/s. The temperature and internal velocity fields change with time as shown in Figure 4. When the inlet cold water penetrates a certain height in the water tank, it falls back. The falling fluid sucks hot water from its adjacent hot water area, forcing the cooler fluid to move down to the bottom of the solar water tank gradually. At the same time, the hot water in the water tank is discharged through the tank outlet. However, with the increase of time, the temperature difference between the cold water and the hot water decreases.

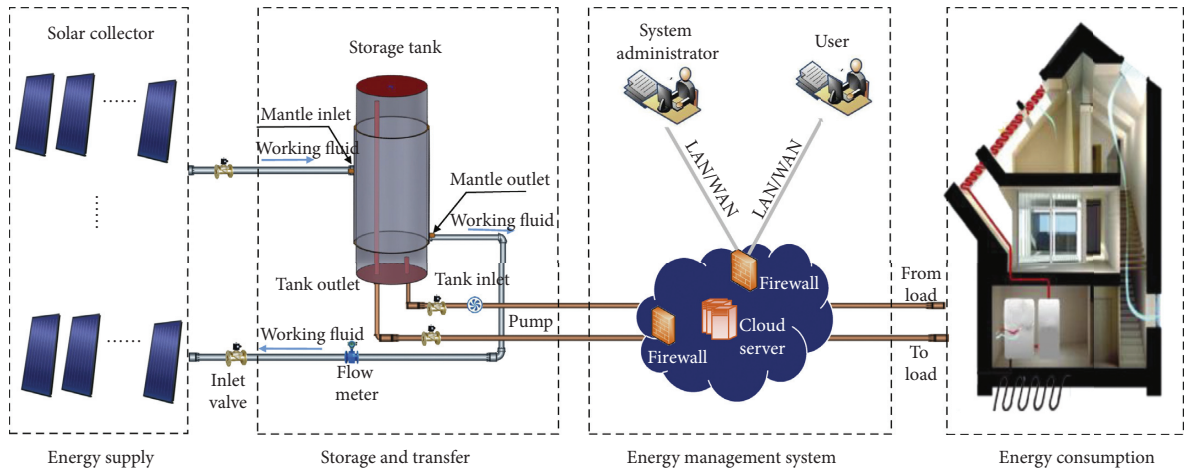

Figure 3: The intelligent solar heating control system based on EIoT.

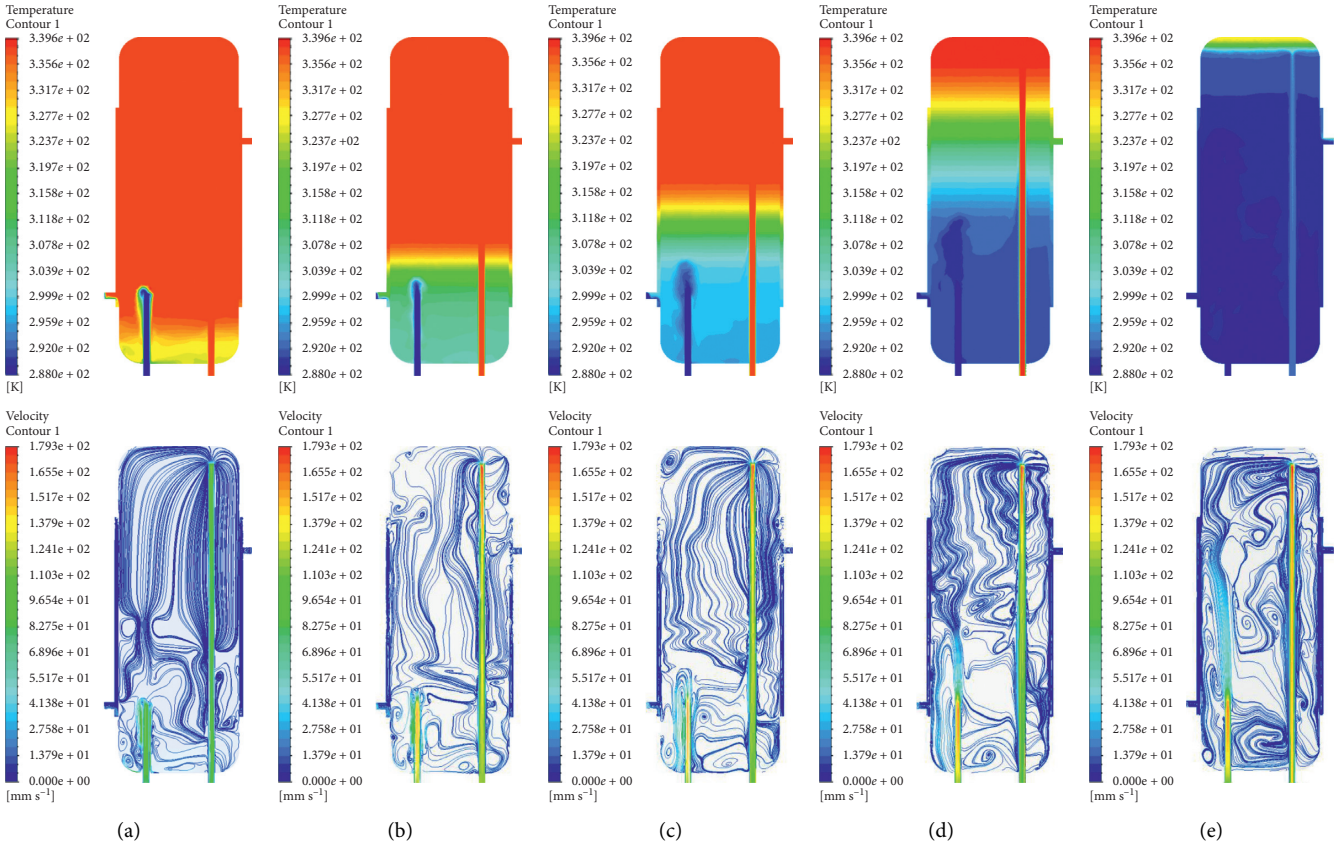

Figure 4: Temperature contours (top row) and velocity streamline (bottom row) in solar water storage tank in the discharging mode. (a) (t) = 100 s, (b) (t) = 900 s, (c) (t) = 1800 s, (d) (t) = 2700 s, and (e) (t) = 3600 s.

After we monitored the temperature of the water outlet of our experimental platform, we obtained a set of temperature records at different times under the same water flow conditions, including 909 sets of abnormal data and 230 sets of average data. Here, we classify the data based on the body's tolerance to water temperature during bathing: 310.15 (K) ~ 315.15 (K). All temperatures data within the range are classified as standard data, and data outside this temperature range are classified as abnormal data. The above is our design philosophy for the temperature monitoring system of the Internet of Things. However, if this temperature monitoring system is attacked by the network, the following results will be produced: the failure of the temperature alarm system will

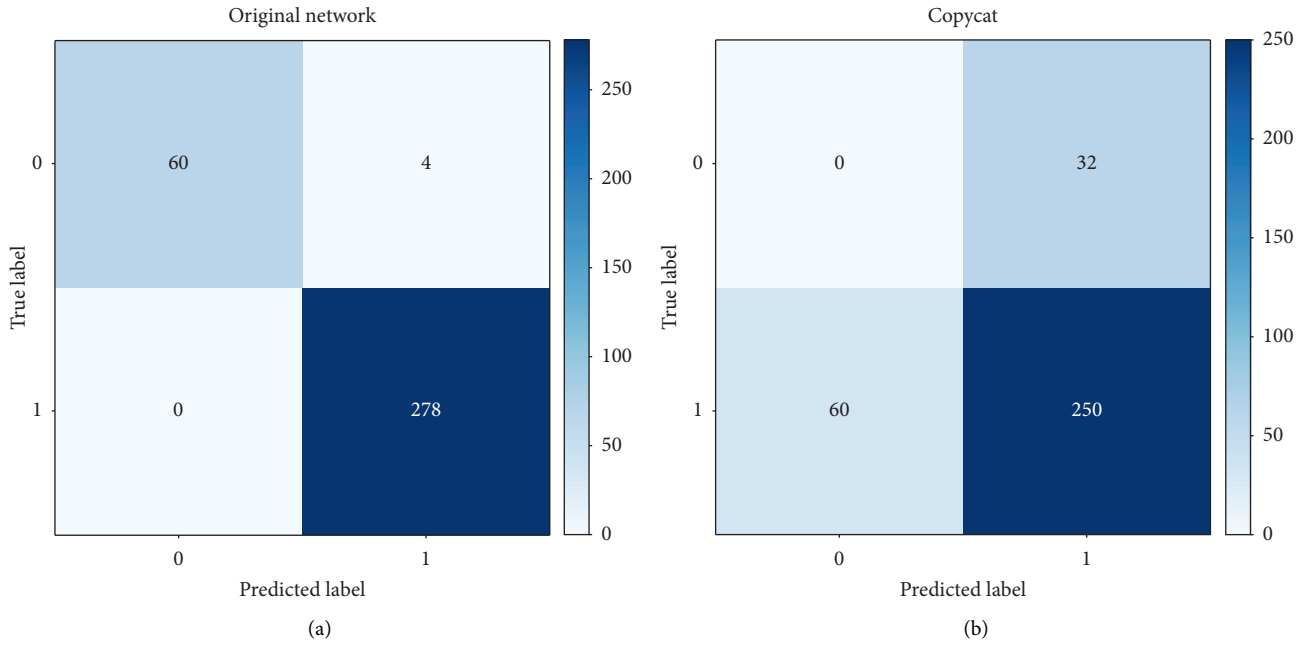

Figure 5: The confusion matrix for average temperature and abnormal temperature prediction.

threaten the personal safety of users. The theft of the entire network of the temperature control system will cause direct economic losses to the owner. And the whole copycat network we designed is based on stealing the whole of RandomForest prediction system to warn our system.

4.2. Performance of the Temperature Control System. The confusion matrix is generally used to evaluate the network output. The following is the definition of confusion matrix. Confounding matrix is a situation analysis table that summarizes the prediction results of classification models in machine learning. In the form of matrix, records in the dataset are summarized according to two criteria of real classification and classification judgment predicted by classification models. TP (True Positive), FN (False Negative), FP (False Positive), and TN (True Negative) are the four elements of the confusion matrix that reflect the performance of the model [24]. A high proportion of TP means that the performance of the entire network is satisfied.

In the predictive classification model, the larger the number of TP and TN, and the smaller the number of FP and FN, the higher the prediction accuracy (as can be seen from Figure 5). However, the calculations in the confusion matrix are numbers. Sometimes, in the face of large amounts of data, it is difficult to measure the number of models by counting. Therefore, the confusion matrix is an extension of the secondary and tertiary indicators (the lowest indicator addition, subtraction, multiplication, and division) in the basic statistical results [25].

Table 1: Values of different indicators based on the source model.

Object	Precision	Recall	F1-score
Normal T	0.94	1.00	0.97
Abnormal T	1.00	0.99	0.99
Macro avg	0.97	0.99	0.98
Weighted avg	0.99	0.99	0.99
Accuracy	—	—	0.99

$$
\begin{aligned}
Acc &= \frac{tp + tn}{tp + fp + fn + tn},\\
\text{Rec} &= \frac{tp}{tp + tn},\\
\text{pre} &= \frac{tp}{tp + tn},\\
f_1 &= 2^{*}\frac{\text{Pre}^{*}\text{Rec}}{\text{Pre} + \text{Rec}}.
\end{aligned}
\tag{6}
$$

We analyzed the confusion matrix obtained by RandomForest to classify the existing temperature dataset and found that TP and TN accounted for the highest proportion of the total output (278 + 60 = 338, the total of which is 342). Based on this data, we can draw a conclusion: using RandomForest to create a temperature control system can get a high accuracy.

Macro average refers to averaging the recall rates of category 1 and category 0. The weighted average is calculated using the proportion of the sample as the weight. It can be seen from the above table that our model has high prediction accuracy. As can be seen from Table 1, our model has reached a very high accuracy.

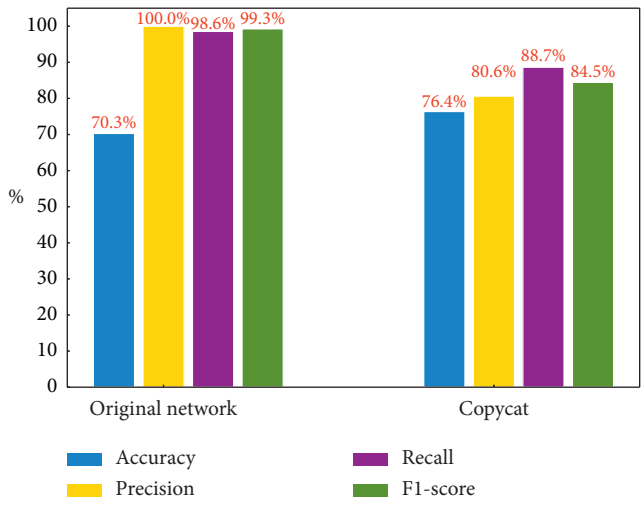

Figure 6: Comparison of different results between the original prediction model and copycat model.

Table 2: Values of different indicators based on a copycat model.

Object(copycat)	Precision	Recall	F1_score
Abnormal T	0.81	0.89	0.84
Macro avg	0.40	0.44	0.42
Weighted avg	0.66	0.73	0.70
Accuracy	—	—	0.73

4.3. Effectiveness of Model Stealing Attack. A RandomForest was trained to predict the average temperature and abnormal temperature. Based on the understanding of temperature data, we set the maximum depth of the (decision) tree in RandomForest to 30. In some other hyperparameters, min_samples_leaf is set to 3, min_samples_split is set to 4, n_estimators is set to 8000, and the values of verbose and n_jobs are both set to 1 [26]. The implementation is based on Pytorch and uses NVIDIA GTX 1070 GPU.

The difference between the original network and the copycat network is fully reflected in Figure 6. Although the precision, recall, and F1-score values obtained by the copycat network are about 15% lower than those of the original network, the accuracy obtained by the copycat network is 6.1% higher than that of the original network, which is enough to prove that the copycat network can pose a threat to the original network.

Based on the data in Table 2, we can conclude that the model that uses machine learning to classify text data can be stolen by different kinds of machine learning. It can be seen from Table 1 that, without considering the average temperature output, the model we used XGBoost to steal can be able to distinguish abnormal temperature data. So in terms of classifying abnormal temperature, the model we stole can already achieve this function.

5. Conclusions

A solar water temperature monitoring system based on the Internet of Things was established in this article. Based on the temperature monitoring system, we propose a RandomForest for normal temperature and abnormal temperature classification. In order to demonstrate the attack on the established model on the IoT platform, an XGBoost model was built by using a small number of labeled samples to steal the known target model. Experimental results show that the replication model can successfully replicate the performance of the target RandomForest, with small performance differences. The success of this attack shows that intellectual property rights such as artificial intelligence models similar to temperature monitoring systems that have been successfully established can be stolen. How to effectively solve these problems has become an urgent problem in the field of deep learning.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (62062069, 62062070, and 62005235).

References

[1] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial internet of things architecture: an energy-efficient perspective," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48–54, 2016.

[2] Y. Kabalci, E. Kabalci, S. Padmanaban, J. B. Holm-Nielsen, and F. Blaabjerg, "Internet of things applications as energy internet in smart grids and smart environments," *Electronics*, vol. 8, no. 9, p. 972, 2019.

[3] S. O. Muhanji, A. E. Flint, and A. M. Farid, *Transactive Energy Applications of eIoT: The Development of the Energy Internet of Things in Energy Infrastructure*, eIoT, Berlin, Germany, 2019.

[4] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the internet of things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, 2017.

[5] A. S. Sani, D. Yuan, J. Jin et al., "Cyber security framework for internet of things-based energy internet," *Future Generation Computer Systems*, vol. 93, pp. 849–859, 2018.

[6] X. Xiaohui, "Study on security problems and key technologies of the internet of things," in *Proceedings of the 2013 International Conference on Computational and Information Sciences*, pp. 407–410, IEEE, Hubei, China, June 2013.

[7] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[8] H. Lin and N. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[9] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot security (IoTsec) mechanisms for e-health and ambient assisted living

applications," in *Proceedings of the 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 13–18, IEEE, Philadelphia, PA, USA, August 2017.

[10] T. Baranwal and P. K. Pateriya, "Development of IoT based smart security and monitoring devices for agriculture. 2016 6th international conference-cloud system and big data engineering (confluence)," in *Proceedings of the 6th Conference on Cloud System and Big Data Engineering*, pp. 597–602, IEEE, Noida, India, January 2016.

[11] M. Nawir, A. Amir, N. Yaakob et al., "Internet of things (IoT): taxonomy of security attacks," in *Proceedings of the 2016, 3rd bInternational Conference on Electronic Design (ICED)*, IEEE, Phuket, Thailand, August 2016.

[12] F. Hussain, R. Hussain, S. A. Hassan et al., "Machine learning in IoT security: current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 99, 2020.

[13] S. Zeadally and M. Tsikerdekis, "Securing internet of things (IoT) with machine learning," *International Journal of Communication Systems*, vol. 33, no. 1, Article ID e4169, 2020.

[14] R. Gomes, M. Ahsan, and A. Denton, "Random forest classifier in SDN framework for user-based indoor localization," in *Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 0537–0542, IEEE, Rochester, MI, USA, May 2018.

[15] M. Ruiz-Abellón, A. Gabaldón, and A. Guillamón, "Load forecasting for a campus university using ensemble methods based on regression trees," *Energies*, vol. 11, no. 8, p. 2038, 2018.

[16] N. Manju, B. S. Harish, and V. Prajwal, "Ensemble feature selection and classification of internet traffic using XGBoost classifier," *International Journal of Computer Network & Information Security*, vol. 11, no. 7, 2019.

[17] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287–2310, 2020.

[18] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, "Security and privacy in 6G networks: new areas and new challenges," *Digital Communications and Networks*, vol. 6, 2020.

[19] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.

[20] N. Dalvi, P. Domingos, S. Sanghai et al., "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 99–108, Washington, DC, USA, July 2004.

[21] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 641–647, San Diego, CA, USA, August 2005.

[22] G. Lin, S. Wen, Q.-L. Han, J. Zhang, and X. Yang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, 2020.

[23] M. Barreno, B. Nelson, R. Sears et al., "Can machine learning be secure," in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, pp. 16–25, Taipei, Taiwan, March 2006.

[24] N. Papernot, P. McDaniel, I. Goodfellow et al., "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, Abu Dhabi, UAE, April 2017.

[25] L. Huang, A. D. Joseph, B. Nelson et al., "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, pp. 43–58, Abu Dhabi, UAE, June 2011.

[26] I. M. Bapiyev, B. H. Aitchanov, I. A. Tereikovskyi et al., "Deep neural networks in cyber attack detection systems[J]," *International Journal of Civil Engineering and Technology (IJCIET)*, vol. 8, no. 11, pp. 1086–1092, 2017.